REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
16-03-2011	Final Technical	15-03-2008 - 30-11-2011

4. TITLE AND SUBTITLE
FAST IMPLICIT METHODS FOR STIFF MOVING INTERFACES

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-08-1-0131

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
John Strain

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of California at Berkeley
Department of Mathematics
970 Evans Hall #3840
Berkeley, CA 94720-3840

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
875 N. Randolph St.
Suite 325
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2012-0238

12. DISTRIBUTION/AVAILABILITY STATEMENT
A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The research supported by this grant aimed to solve moving interface problems with two flexible computational modules: semi-Lagrangian contouring (SLC) methods which evolved an interface with a given velocity field, and problem-dependent evaluation of the velocity of a given interface. These modules, with associated engines for fast computational geometry, constituted a highly effective toolbox for evolving an implicitly represented interface through arbitrary topology under a given velocity functional. The first module was extended with linearly-implicit SLC methods which evolved stiff interfaces under geometric velocities such as mean curvature and surface diffusion,and fourth-order SLC methods which delivered greatly improved accuracy. The second module was extended with fast solvers for elliptic systems which evaluated interface velocities determined by quasistationary physical interactions off the interface, and enabled the solution of key models such as Ostwald ripening, elastic membranes in Stokes flow, and crystal growth.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*

Reset

FAST IMPLICIT METHODS FOR STIFF MOVING INTERFACES

AFOSR GRANT FA9550-08-1-0131

John Strain
Department of Mathematics
University of California

Abstract The research supported by this grant aimed to solve moving interface problems with two flexible computational modules: semi-Lagrangian contouring (SLC) methods which evolved an interface with a given velocity field, and problem-dependent evaluation of the velocity of a given interface. These modules, with associated engines for fast computational geometry, constituted a highly effective toolbox for evolving an implicitly represented interface through arbitrary topology under a given velocity functional. The first module was extended with linearly-implicit SLC methods which evolved stiff interfaces under geometric velocities such as mean curvature and surface diffusion, and fourth-order SLC methods which delivered greatly improved accuracy. The second module was extended with fast solvers for elliptic systems which evaluated interface velocities determined by quasistationary physical interactions off the interface, and enabled the solution of key models such as Ostwald ripening, elastic membranes in Stokes flow, and crystal growth.

Research results

A moving interface is a collection $\Gamma(t)$ of nonintersecting oriented closed curves or surfaces with an outward unit normal N, curvature C, and normal velocity V, specified as a functional of $\Gamma(t)$. Examples include passive transport where $V = F(x,t)$ is given; geometric motion $V = (p(N) - q(N)C)N$; Ostwald ripening $V = \left[\frac{\partial u}{\partial N}\right] N = \Lambda C$, where $\Delta u = 0$ off $\Gamma(t)$, $u = C$ on $\Gamma(t)$, and Λ is the Dirichlet-Neumann operator; and models for crystal growth $V = \left[\frac{\partial u}{\partial N}\right] N$, where $u_t = \Delta u$ off $\Gamma(t)$ and $u = f(N, C, V)$ on $\Gamma(t)$. We reformulate moving interface problems as evolution of an implicit representation function φ whose zero set is $\Gamma(t)$, plus some densities defined on $\Gamma(t)$. Then at a new time level $t+k$, the new implicit representation is given by $\varphi(x, t+k) = \varphi(\tilde{x}, t)$ where $\tilde{x}$ is the foot of the characteristic ending at x.

Semi-Lagrangian contouring (SLC) is reviewed in §1.1, applied to Stokes flow in §1.2, and to 3D viscoelastic flow in §1.3. Two results in fast computational geometry are reported in §1.4, and employed in §1.5 for fourth-order SLC. Linearly-implicit SLC for stiff problems is presented in §1.6.

1.1. Semi-Lagrangian contouring Semi-Lagrangian contouring (SLC) delivers a coherent and widely usable black–box module, which moves an interface through arbitrary topological and geometric events, with minimal information about the specific physical problem and the interface velocity V. It incorporates implicit interface evolution, semi-Lagrangian advection, fast geometric algorithms and fast elliptic solvers.

Each point of a moving interface Γ travels along a characteristic curve of the interface velocity V, carrying a zero value of the signed distance

$$d(x) = \pm \min_{\gamma \in \Gamma} \|x - \gamma\|.$$

We extend the velocity V smoothly to a global field W, move all values of d along approximate characteristics of W, and extract the resulting interface by contouring semi-Lagrangian approximate advection formulas such as the first-order Courant-Isaacson-Rees (CIR) scheme

$$\varphi(x) = d(x, t + k) \approx d(x + kV(x, t), t). \tag{1}$$

The analogous second-order scheme couples the first-order predictor (1) with a trapezoidal corrector based on the averaged velocity. It combines the unconditional stability of the first-order scheme with the dramatically reduced dissipation of the trapezoidal rule. Exact distance finding in a dynamic quadtree data structure or an adapted Voronoi diagram eliminates the usual semi-Lagrangian interpolation error. The interface velocity is extended by a numerical Whitney extension, which satisfies a maximum principle and removes the discontinuities of the usual nearest-point extension. After evolving the implicit representation φ, the new interface is extracted by subgrid refinement of a tree mesh: given a signed distance function $\varphi(x, t_n)$, we build a tree at time $t_{n+1} = t_n + k$ by recursive evaluation of $g(x) = \varphi(s, t_n)$ at projected points $s = x + kW(x, t_n)$. SLC methods converge to correct viscosity solutions for difficult moving interface problems involving merging, faceting, transport, nonlocality and anisotropic curvature-dependent geometry [12].

1.2. Stokes flow with elastic interfaces Slow viscous flows which satisfy the incompressible Stokes equations

$$-\nu \Delta u + \nabla p = F, \qquad \nabla \cdot u = 0$$

commonly occur in biological moving interface problems, for example blood flow, cell movement, and atherosclerosis. The force F in these equations is often modeled by a measure $F = f \delta_\Gamma$ when the interface Γ has a complex internal elastic structure. We have developed a fast solver for Stokes flow with elastic interfaces, by combining semi-Lagrangian interface evolution with a fast new Ewald summation scheme [3]. The semi-Lagrangian transport of interface densities allows a remarkably straightforward computation of stretching energy, while the new fast summation technique unifies

several well-known local correction techniques for singular integral operators. Our technique remains highly effective with discontinuous data, where standard Stokes solvers encounter substantial difficulty. Figure 1 exhibits circularizing and oscillating interfaces under Stokes flow, accurate to one part per thousand.

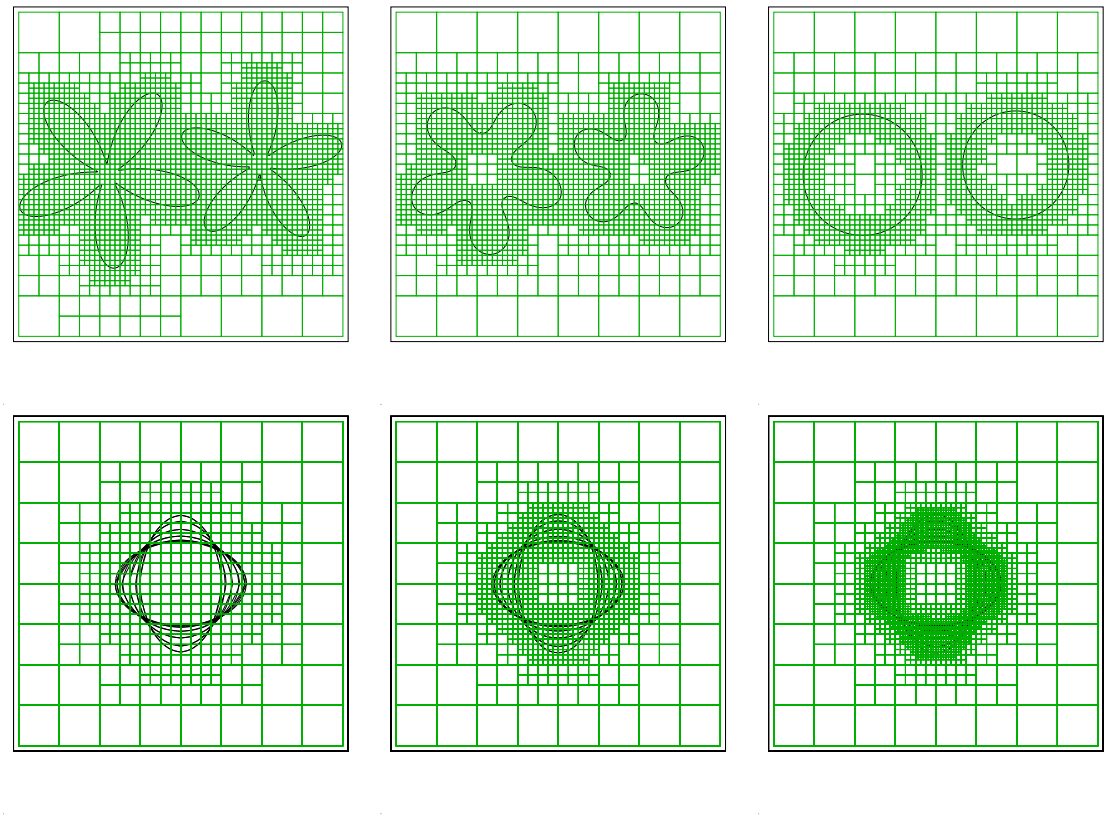


Figure 1: Circularizing and oscillating interfaces in Stokes flow.

1.3. 3D viscoelastic flow with structured interfaces The semi-Lagrangian method has been implemented in 3D and coupled with a viscoelastic fluid simulator to produce complex and realistic fluid simulations [1, 2]. It is publicly available as part of the open-source Berkeley Fluid Animation and Simulation Toolkit (BFAST) on SourceForge. The semi-Lagrangian approach is capable of tracking local features such as the surface colors shown in Figure 2 or the elastic energy density of a tensile membrane [3].

1.4. Computational geometry Operations on implicit representations are widely useful in computational science. We are extending our efficient 2D piecewise-linear modules for implicit geometric operations such as contouring, distancing and extension, to 3D higher-order piecewise-polynomial interfaces.

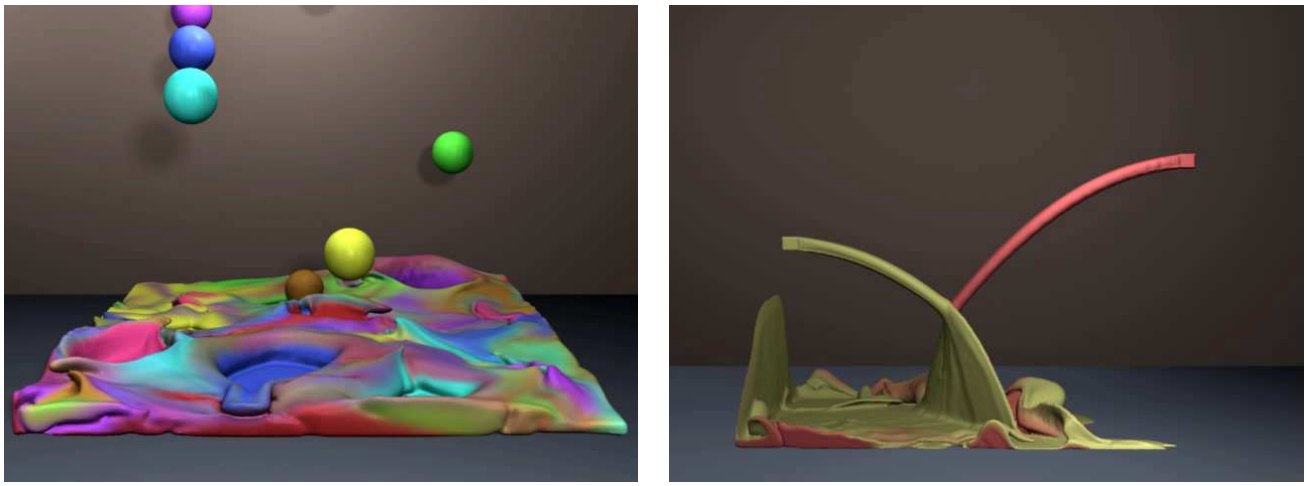


Figure 2: Paint balls and merging sprays in 3D viscoelastic flow.

Accurate contouring The general contouring problem, finding a smooth geometrically constrained approximate zero set of a function which can be evaluated at arbitrary points, occurs frequently in computational science and requires a robust package. An ideal contouring package would accept function values (and derivatives if available) at arbitrary points, and produce a piecewise-smooth approximation to the zero set, with corners where necessary. Geometric constraints, such as bounds on curvature away from corners, are vital in applications like computer-controlled machining. They pose a major complication for existing public-domain contouring software. We are finalizing new C/C++ packages for constrained piecewise-smooth contouring in two and three dimensions, based on new local contouring schemes for Bezier patches and new methods of scattered data interpolation [6, 7].

Fast distance to Bezier curves Computation of distances to an explicit surface is a key step in implicitization, and extremely expensive if implemented directly. Fast distancing algorithms are available for standard piecewise-linear polygonal surfaces. Smoother piecewise-polynomial Bezier surfaces would provide higher accuracy, but computing distances to these surfaces has been intractable. The classical Voronoi diagram provides efficient algorithms for distance finding to "sites" which are points in space, but fails when the sites are Bezier segments of a piecewise-smooth surface. We have developed a fast Voronoi-based distance finding algorithm for Bezier segments, which dramatically speeds up the computation of distance to piecewise-polynomial curves and surfaces [5]. Figure 3 demonstrates the speed and robustness of our algorithm, which requires no more than five elementary distance finding steps per evaluation.

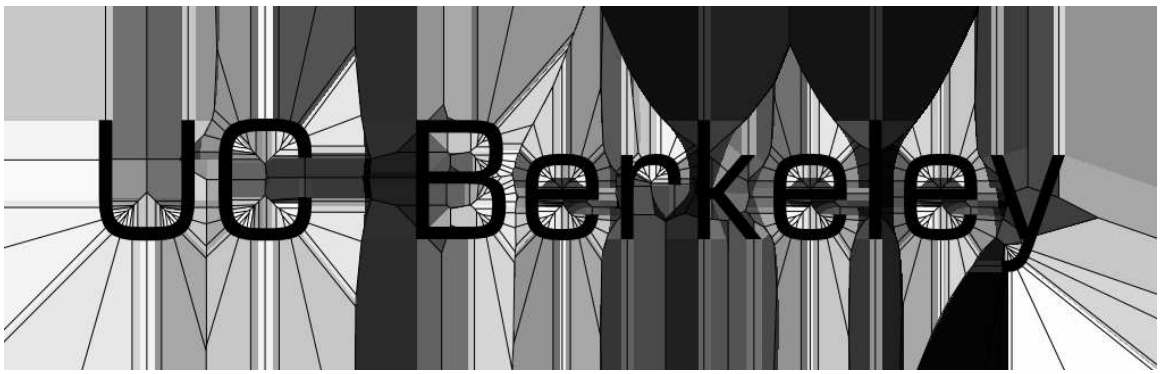


Figure 3: Fast distancing to piecewise-polynomial curves: the background shading from white to black indicates regions where the algorithm requires 1 to 5 elementary distancing operations respectively.

1.5. Fourth-order accurate SLC methods New SLC methods based on piecewise-cubic contouring, fast distancing and fourth-order time discretization dramatically reduce the number of degrees of freedom required to represent complex nearly-singular interfaces (see Fig. 4)[8, 10].

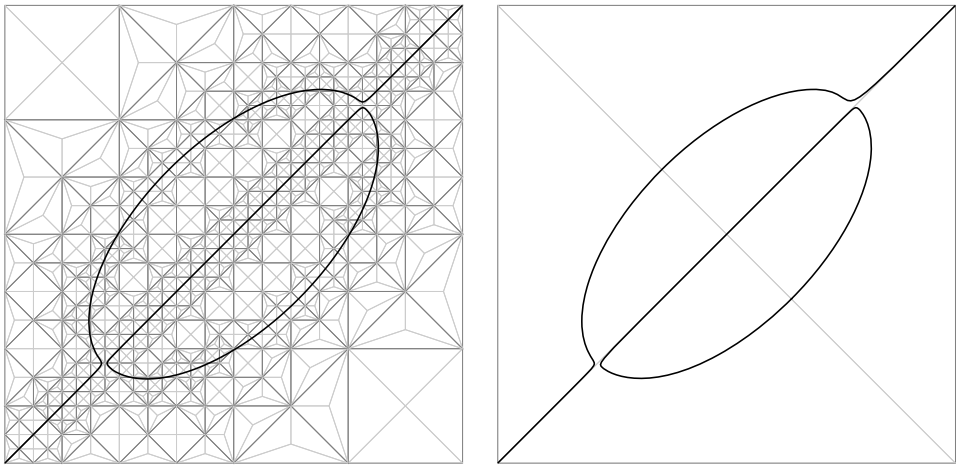


Figure 4: Fourth-order cubic approximation of a nearly singular contour (right) is far more efficient than second-order linear approximation (left).

Fourth-order time discretization integrates the characteristic terminal value problem

$$x'(s) = W(x(s), s) \qquad x(t+k) = x$$

backward in time till $s = t$. Input data consists of $x(t+k) = x$ at the terminal time $t+k$, and $W(x,t)$ at the initial time t. Subsequent values of $W(x,s)$ become available only after each new approximation of $x(s)$ has been computed. The mismatch in time

between x and W requires unconventional time discretization schemes such as the first-order CIR predictor

$$x_0 = x, \quad W_{10} = W(x,t), \quad x_1 = x - kW_{10}$$

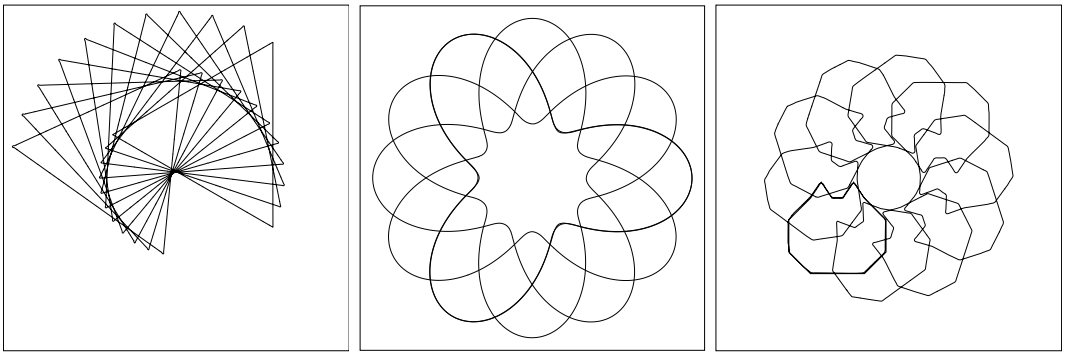
and the second-order trapezoidal corrector

$$W_{10} = W(x_1,t), \quad W_{01} = W(x_0,t+k), \quad x_2 = x - \frac{k}{2}W_{01} - \frac{k}{2}W_{10}.$$

Thus we derive higher-order correctors

$$W_{20} = W(x_2,t), \qquad x_3 = x - \alpha k W_{20} - \beta k W_{10} - \gamma k W_{01}$$

by the method of undetermined coefficients [10]. Numerical computations of passive rotation and geometric evolution demonstrate the efficiency of fourth-order SLC for moving complicated smooth interfaces under nonstiff velocity fields (Fig. 5).



(a) (b) (c)

Figure 5: High-order semi-Lagrangian contouring evolves smooth and nonsmooth shapes accurately under smooth geometric velocity fields.

1.6. Stiff problems The interface velocity V often depends on the moving interface $\Gamma(t)$ in a stiffly-stable way. Local geometric problems such as mean curvature flow $V = -CN$ and surface diffusion $V = (\Delta C)N$ are equivalent to stiff nonlinear parabolic PDEs for the implicit interface representation φ. Nonlocal problems of materials science (crystallization, Ostwald ripening, Hele Shaw flow) determine the interface velocity from a global system of PDEs with stiff geometric boundary conditions. Our new stiffly-stable implicit SLC methods and damped Newton-Krylov solvers provide efficient solution methods for stiff moving interface problems.

For stiff moving interface problems, standard explicit formulas such as first-order CIR scheme

$$g(x) := \varphi(x, t+k) = \varphi(x - kW(x,t), t)$$

are stable in the maximum norm but not in norms involving higher derivatives. The resulting interface oscillations can be stabilized by taking small time steps or frequent smoothing.

Efficient solution of stiff problems requires implicit time discretizations such as the first-order implicit formula

$$g(x) = \varphi(x - E_g V_g(x, t+k), t) =: F[g](x)$$

in which the known velocity $W(x,t)$ is replaced by the unknown velocity $W(x,t+k) = E_g V_g(x,t+k)$. The velocity $W(x,t+k)$ depends on the unknown implicit interface representation $g(x) \approx \varphi(x,t+k)$ twice. The operator E_g extends functions from the zero set Γ of g to the whole ambient space, while the problem-dependent interfacial velocity V_g depends stiffly on derivatives of g. We solve the implicit interface motion by damped Newton iteration and Krylov space methods, and obtain excellent results for mean curvature flow. Nonlocal problems are treated by the chain rule and a library of basic Frechet derivatives.

Damped Newton iteration Since V_g usually depends on geometric derivatives of g such as

$$N = \frac{\nabla g}{\|\nabla g\|}, \qquad C = \nabla \cdot N,$$

the simple fixed point iteration $g \leftarrow g - F[g]$ will not converge. Thus we employ damped Newton iteration

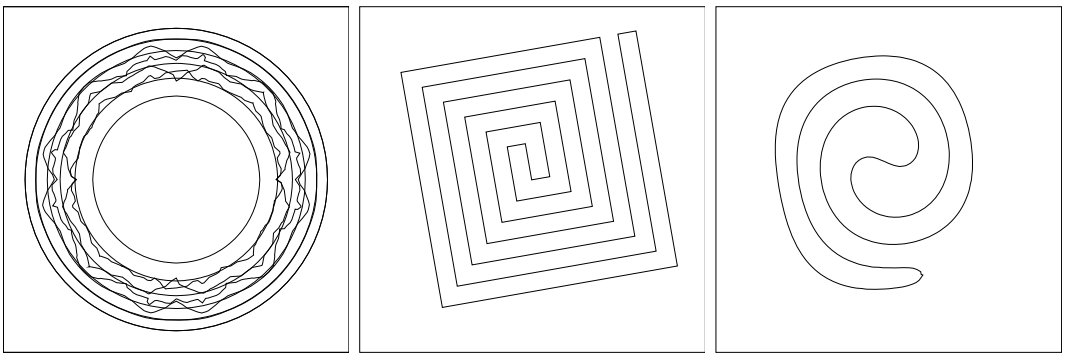
$$DF[g]h = -F[g], \qquad g \leftarrow g + \lambda h.$$

Here $DF[g]$ is the Frechet derivative of F at g. The damping parameter λ is adaptively chosen to guarantee steady decrease of the residual: $\|F[g]\| \leftarrow \|F[g]\|/2$. Damped Newton iteration preconditions the loss of derivatives in each F evaluation by the inverse Jacobian $DF[g]^{-1}$, yielding a modified fixed point iteration with improved convergence.

At each damped Newton step, the variational equation $DF[g]h = -F[g]$ is approximated by a large sparse system of linear equations and solved by linear or nonlinear Krylov space methods. Linear methods such as GMRES converge faster, but each step requires an expensive evaluation of the Jacobian DF. Nonlinear methods require only F evaluations.

Implicit SLC with damped Newton iteration, solved by Krylov space methods, deliver efficient algorithms for stiff geometric moving interface problems [8, 11]. Numerical results for mean curvature flows of smooth and nonsmooth objects are shown in Fig. 6.

Chain rule The Frechet derivative $D_\varphi V$ involved in second-order elliptic moving interface problems (such as Ostwald ripening $V = \Lambda_\varphi C$) is a problem-dependent



(a) (b) (c)

Figure 6: Implicit semi-Lagrangian contouring evolves both circles (superimposed on unstable explicit results) and nonsmooth shapes stably under mean curvature flow.

third-order linear pseudodifferential operator. It can be constructed from problem-independent components by the chain rule:

$$D_\varphi V = D_\varphi(\Lambda_\varphi C) = D_\varphi \Lambda_\varphi(C) + \Lambda_\varphi(D_\varphi C).$$

Hadamard's variational formula implies an augmented elliptic system for the first component $D_\varphi \Lambda_\varphi$, derived by implicit differentiation of the PDEs and boundary conditions which define Λ_φ. The second component $D_\varphi C$ is a problem-independent local geometric sensitivity. Efficient problem-independent modules can compute and assemble these components of the Frechet derivative for a wide variety of stiff moving interface problems.

2. Fast elliptic solvers Elliptic partial differential equations (PDEs) for the interface velocity are converted to first-order systems in §2.1 and solved by finite differences in §2.2. New Alternating Direction Implicit (ADI) iterations for elliptic systems are presented in §2.3, and generalized Ewald summation for elliptic systems in §2.4. Two nonuniform fast Fourier Transforms for piecewise-polynomial distributions are described in §2.5 and §2.6, and employed in §2.7 and §2.8, to speed up locally-corrected boundary integral solution of elliptic systems in complex domains.

2.1. First-order systems Industrial moving interface problems (such as solidification, fluid-structure interaction and crystal growth) determine the interface velocity V by solving systems of PDEs on the moving phase domains, with boundary conditions on fixed and moving boundaries. Moving interfaces with our SLC methods relies on a problem-dependent module which evaluates the interface velocity V of the interface $\Gamma = \{\varphi = 0\}$. Implicit time discretization of stiff problems requires the Frechet derivative $D_\varphi V$ with respect to the interface representation φ, which satisfies additional PDEs. New locally-corrected spectral boundary integral methods

[13, 17, 16] solve general elliptic systems of PDEs with complex interfaces: Linear constant-coefficient elliptic systems (such as Maxwell, Stokes, or linear elasticity) are converted to overdetermined first-order systems

$$Au(x) = \sum_{j=1}^{d} A_j u_{,j}(x) + A_0 u(x) = f(x) \text{ in } \Omega, \qquad Bu = g \text{ on } \Gamma = \partial\Omega \tag{2}$$

where each A_j is a $p \times q$ matrix, $B = B(\gamma)$ is an $r \times q$ matrix for each $\gamma \in \Gamma$, and u is a q-vector. Such systems are amenable to new iterations and simple new boundary integral solvers.

2.2. Finite difference methods A new piecewise-polynomial interface method for discretizing elliptic systems with complex interfaces between high-contrast materials is derived and analyzed in [4]. A Krylov-accelerated interface multigrid approach solves the new discretization efficiently. Stability and convergence are proved in one dimension. Numerical experiments with complex two-dimensional interfaces (see Fig. 7) and coefficients varying over eight orders of magnitude demonstrate the accuracy, efficiency and robustness of the method.

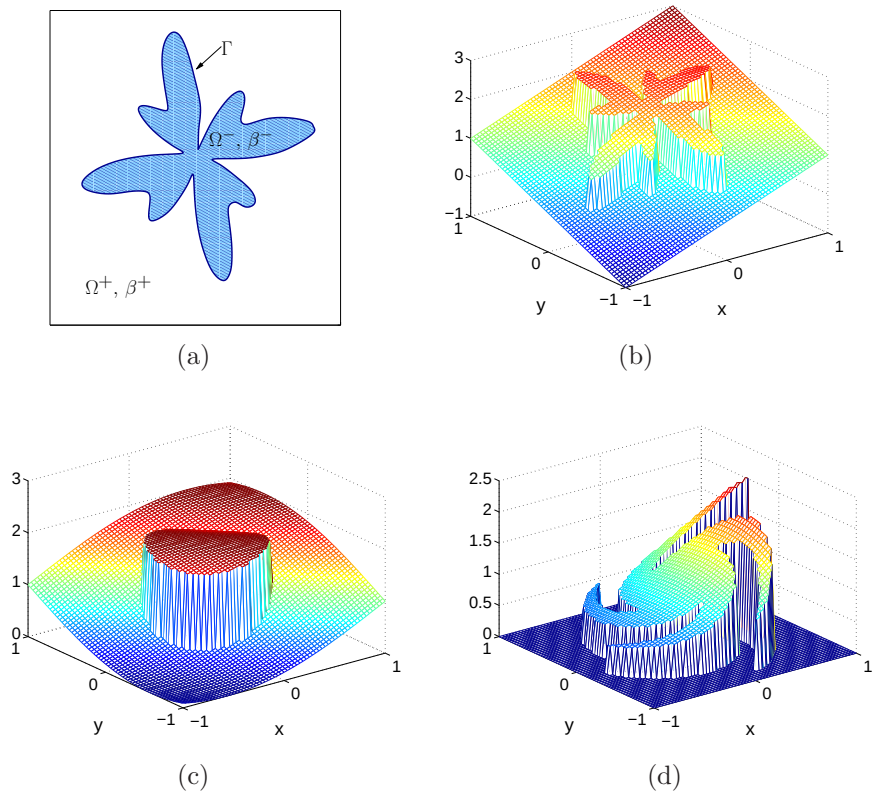


Figure 7: Solution of elliptic problems with complex interfaces.

2.3. ADI methods for elliptic systems Classical ADI methods provide the oldest essentially optimal iteration, for separable second-order positive definite elliptic problems

$$Au + Bu = -\partial_x (\alpha \partial_x u) - \partial_y (\beta \partial_y u) = f,$$

in rectangular two-dimensional domains $\Omega \subset R^2$. With scale $1/s$, these iterations take the form

$$\left(s^2 + A\right)\left(s^2 + B\right) u^{m+1} = \left(s^2 - A\right)\left(s^2 - B\right) u^m + 2f$$

and modify each Fourier mode $\mathrm{e}^{\mathrm{i}k^T x}$ with wavenumber $k = (k_1, k_2)$ by a symbol σ which damps error rapidly over a geometric range of wavenumbers. For example, in the Poisson equation $|\sigma(k)| \leq \frac{1}{3}$ whenever $\frac{1}{\sqrt{2}} \leq \frac{|k|}{s} \leq \sqrt{2}$. Thus varying the scale s geometrically gives $O(\epsilon)$ damping in $O(\log N \log \epsilon)$ sweeps. Each sweep approximates $(A + B)^{-1}$ by a product of one-dimensional operators $(s^2 \pm X)^{\pm 1}$ and requires time proportional to the problem size, making ADI an essentially optimal iteration.

We derive ADI methods for general elliptic systems (2). Ellipticity implies that in any bounded domain Ω with boundary $\Gamma = \partial\Omega$ and inward normal vector field n, the inward normal derivative of u is determined by values and tangential derivatives of u on Γ:

$$\partial_n u = \sum_i n_i \partial_i u = A_n^\dagger \left(f - A_T \partial_T u - A_0 u\right). \tag{3}$$

Here the left inverse $A_n^\dagger$ of $A_n = \sum_{j=1}^d n_j A_j$ exists by ellipticity, and $A_T \partial_T$ is the tangential part of the operator $A = A_n \partial_n + A_T \partial_T$. Given the inward normal derivative, the solution can be approximated by Taylor expansion on a strip near the boundary. Marching inward with repeated Taylor expansion solves the boundary value problem, and explains heuristically why boundary value problems are well-posed for elliptic systems. Thus our ADI methods sweep inward from the boundary to provide convergent three-step iterations for elliptic systems: 1. Choose sweep direction n pointing in from the boundary and rewrite the system in normal/tangential coordinates (3). 2. Left-invert A_n and shift by the inverse length scale s to get the equivalent system

$$su + \partial_n u + B_0^+ u = su + f - B_T \partial_T u + B_0^- u. \tag{4}$$

3. Solve Eq. (4) inward from the boundary across the domain, with step size proportional to $1/s$. Repeating these three steps with a sequence of directions n and geometrically varying scales s gives a full ADI iteration.

Convergence In rectangular domains with periodic boundary conditions, a collection of sweeps with different directions n reduces each error mode $\mathrm{e}^{\mathrm{i}k^T x} I$ by a matrix-valued symbol

$$\sigma(k) = \prod_n \left(s + \mathrm{i}k_n + B_0\right)^{-1} \left(s - \mathrm{i}B_T k_T\right).$$

Employing a larger set of directions improves the convergence speed exponentially: one pass of ADI iteration reduces the error by 0.5 with 4 directions, 0.1 with 8 directions, and 0.005 with 16 directions. Since this error reduction is independent of mesh size (see Figure 8), the new ADI iteration is an essentially optimal solver for elliptic systems in simple domains [14].

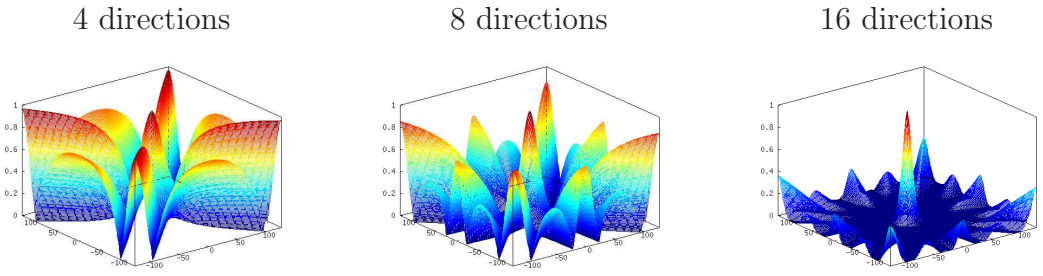


Figure 8: ADI error reduction symbol σ versus wavenumber k for the Poisson equation on a $N = 256^2$ mesh, solved with 4, 8 or 16 directions in $O(N \log N)$ work.

2.4. Generalized Ewald summation For elliptic systems with complex interfaces, locally-corrected spectral methods [13, 17, 16] employ the periodic fundamental solution

$$S(x-y) = \sum_{k \in Z^d} \left(\sum_{j=1}^{d} \mathrm{i} k_j A_j + A_0 \right)^{\dagger} \mathrm{e}^{\mathrm{i} k^T (x-y)} = \sum_{k \in Z^d} a(k)^{\dagger} \mathrm{e}^{\mathrm{i} k^T (x-y)}, \qquad a^{\dagger} = (a^* a)^{-1} a^*$$

of the first-order system (2). Generalized Ewald summation [13] extracts a global rapidly-converging Fourier series

$$S_\tau(x-y) = \sum_{k \in Z^d} \mathrm{e}^{-\tau a^*(k) a(k)} a(k)^{\dagger} \mathrm{e}^{\mathrm{i} k^T (x-y)},$$

mollified by a matrix exponential $\mathrm{e}^{-\tau a^*(k) a(k)}$. The error term has a formal asymptotic expansion as $\tau \to 0$

$$E_\tau = S - S_\tau = (I - \mathrm{e}^{-\tau A^* A})(A^* A)^{-1} A^* = \tau A^* - \frac{\tau^2}{2!}(A^* A) A^* + \cdots \tag{5}$$

which corrects via an asymptotic series of local differential operators. These new mollification and local correction techniques combine with the fast Fourier transform, Padé codes for small dense matrix exponentials, and high-order uncentered differencing to solve first-order elliptic systems in periodic geometry. A simple algebraic algorithm automatically computes local correction coefficients which achieve high-order accuracy at minimal cost.

Locally-corrected spectral volume potentials Fast solvers for elliptic systems (2) employ the volume potential

$$Vf(x) = \int_\Omega S(x-y)f(y)\,dy$$

to eliminate the right-hand side f. Such potentials solve elliptic systems with piecewise-smooth right-hand sides which are discontinuous across the interface $\Gamma = \partial\Omega$. Locally-corrected spectral methods treat the interfacial discontinuities as distributions, converting the problem to an inhomogeneous system $Au = F$ with a singular right-hand side $F = f\chi_\Omega$ on a fixed enclosing domain $B \supset \Omega$. Here f is smooth, while the characteristic function χ_Ω of the domain Ω jumps from 1 to 0 across the interface Γ and includes all the singularities of the data. Generalized Ewald summation localizes and separates the singular interface distributions to solve the inhomogeneous system accurately and efficiently. The Fourier series $V_\tau = S_\tau * F$ is computed by a geometric nonuniform fast Fourier transform (GNUFFT) [9, 15]. The local correction (5) separates smooth from nonsmooth parts of the solution by the product rule for derivatives. For example, the first-order correction $E_\tau = \tau A^* + O(\tau^2)$ yields

$$\tau A^*(f\chi_\Omega) = \tau\left(\sum_{j=1}^{d} A_j^*(f_{,j}\chi_\Omega + f n_j \delta_\Gamma) + A_0^* f\chi_\Omega\right)$$

where n is the outward unit normal to Ω and δ_Γ is a delta-function on Γ. Since the enclosing domain is fixed and simple, the resulting locally-corrected volume potential capitalizes on fast solvers for simple problems to achieve optimal efficiency [17].

2.5. GNUFFT by B-spline smoothing Fast solvers based on Ewald summation require the computation of accurate low-frequency Fourier coefficients of singular distributions such as δ-functions spread over curves, surfaces, and other lower-dimensional geometric objects. Standard nonuniform FFT (NUFFT) algorithms work only for point distributions. New geometric NUFFTs or GNUFFTs compute accurate Fourier coefficients of a piecewise-polynomial distribution supported on arbitrary-dimension simplices scattered in D-dimensional Euclidean space. Each singular term is mollified with a B-spline smoothing kernel, evaluated on a nearby uniform mesh, transformed with a standard FFT, and deconvolved in real space. The resulting algorithm displays dramatic speedups over direct evaluation, reduces to the standard NUFFT in simplex dimension zero, and provides a highly effective tool for the locally-corrected spectral solution of elliptic systems [9].

2.6. GNUFFT via the butterfly algorithm A new GNUFFT based on the butterfly algorithm generalizes pointwise NUFFTs based on low-rank approximations such as the Taylor expansion

$$\mathrm{e}^{t_i s_j} = \mathrm{e}^{t_i \sigma} \sum_{\alpha=0}^{m} \frac{(t_i-\tau)^\alpha (s_j-\sigma)^\alpha}{\alpha!} \mathrm{e}^{-\tau\sigma} \mathrm{e}^{\tau s_j} + O(\rho^m/m!),$$

valid in any Heisenberg rectangle $|t-\tau||s-\sigma| \le \rho$. The butterfly GNUFFT builds moments due to sources s_j in Heisenberg rectangles with minimal span in source space, recursively merges and shifts them to maximize source span and minimize target span, and evaluates the final expansions at targets in optimal $O(N \log N)$ time [15]. The recursion is illustrated in Fig. 9.

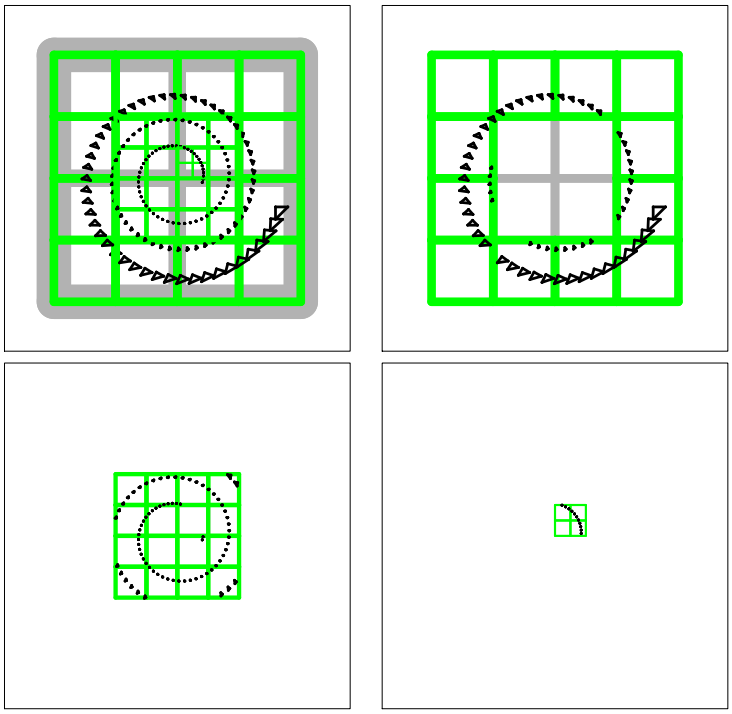


Figure 9: Recursive subdivision of a spiral of simplices for the butterfly GNUFFT.

2.7. Locally-corrected spectral boundary integral solvers A boundary integral equation is derived in [13] with the periodic fundamental solution $S(x-y)$ of the elliptic system (2): multiplying Eq. (2) by S, integrating over Ω and applying Gauss' theorem yields

$$\frac{1}{2}\mu(\gamma) - \int_\Gamma P(\gamma)S(\gamma-\sigma)A_n(\sigma)\mu(\sigma)\,d\sigma = \rho(\gamma), \qquad A_n(\sigma) = \sum_{j=1}^{d} n_j(\sigma)A_j. \quad (6)$$

The new unknown $\mu(\gamma) = P(\gamma)u(\gamma) = (I - B^*(\gamma)B(\gamma))u(\gamma)$ is the projection of u orthogonal to the data g. The right-hand side is a combination of volume and layer potentials

$$\rho(\gamma) = \int_\Omega P(\gamma)S(\gamma-y)f(y)\,dy + \int_\Gamma P(\gamma)S(\gamma-\sigma)A_n(\sigma)B^*g(\sigma)\,d\sigma.$$

Eq. (6) is square, well-conditioned, and amenable to solution by a variety of fast stable high-order accurate methods. The solution u of the elliptic system (2) is recovered locally on the boundary by $u(\gamma) = \mu(\gamma) + B^*(\gamma)g(\gamma)$, and globally by integration.

The boundary integral equation (6) is solved with generalized Ewald summation for S (§2.4), which approximates the kernel $K(\gamma, \sigma) = P(\gamma)S(\gamma - \sigma)A_n(\sigma)$ by a global rapidly-converging Fourier series, and converts the integral equation into semi-separated form

$$(I - MRT)\mu = \rho. \tag{7}$$

Here $T\rho$ evaluates Fourier coefficients of $(A_n\rho)\delta_\Gamma$, R is the filtered inverse of the elliptic operator in Fourier space, and M evaluates and projects Fourier series on Γ. Since $(I - MRT)^{-1} = I + MR(I - TMR)^{-1}T$, the semi-separated form (7) can be solved in Fourier space, where the matrix of TMR is

$$\widehat{\Gamma}(k, q) = 2\int_\Gamma A_n(\sigma)\mathrm{e}^{-\mathrm{i}(k-q)^T\sigma}\,\mathrm{d}\sigma\mathrm{e}^{-\tau a^*(q)a(q)}a^*(q).$$

This matrix contains the elliptic system, interface and boundary conditions. Its regular data structure is amenable to fast randomized low-rank approximation and efficient solution. Local corrections are higher-order in the small parameter τ, and can be applied locally after each Fourier space iteration. The resulting boundary integral solver is efficient, stable, and as accurate as the underlying interface representation.

Frechet derivatives Our boundary integral approach expresses the interface velocity V by a sequence of three explicit differentiable steps which facilitate the approximation of Frechet derivatives. First, the solution u of the elliptic system is a local function

$$u(\gamma) = (I - B^*B)u(\gamma) + B^*g(\gamma) = \mu(\gamma) + B^*g(\gamma)$$

of μ on the interface. Its values off the interface depend only on the known values on the interface. Second, the integral density μ satisfies the boundary integral equation

$$(I - MRT)\mu = \rho,$$

where the operators M and T depend on φ. The damped pseudoinverse R of the elliptic operator A is independent of the interface. Third, the right-hand side ρ of the boundary integral equation depends on φ via integration over the interface and projection by the boundary matrix B. In moving interface problems, g depends on the normal n and curvature C, and $D_\varphi V$ involves the corresponding Frechet derivatives.

Given this three-step construction of the solution u, the standard formula $D(X^{-1}) = -X^{-1}DXX^{-1}$ and the chain rule exhibit $D_\varphi\mu$ as a sum

$$D_\varphi\mu = (I - MRT)^{-1}\left((D_\varphi M)RT + MRD_\varphi T\right)(I - MRT)^{-1}\rho + (I - MRT)^{-1}D_\varphi\rho \tag{8}$$

of products of complicated operators. Each factor encapsulates the linearized response of one component of the system to a perturbation in the interface. Eq. (8) determines the exact Frechet derivative $D_\varphi V$ and generalizes the classical Hadamard formula.

3. Interactions/Transitions

The PI presented results from this research in seminars and colloquia at FAN2010: Fluid dynamics, analysis and numerics (Duke University), the IMA Hot Topics Workshop on Integral Equation Methods, Fast Algorithms and Applications (Minneapolis), the International Conference on Industrial and Applied Mathematics (Zurich and Vancouver), the International Conference on Advances in Scientific Computing (Brown University), California Institute of Technology, Duke University, Michigan State University, the National University of Singapore, North Carolina State University, Purdue University, University of North Carolina at Charlotte, University of California at Berkeley, University of California at Irvine, the American Institute of Mathematics, the Institute for Mathematics and its Applications, INRIA Paris-Rocquencourt, and the Statistical and Applied Mathematical Sciences Institute.

Efficient and accurate 2D contouring and distancing codes are nearing completion, and will soon be open-sourced from the PI's website [5, 7]. High-order implicit semi-Lagrangian contouring codes for moving interfaces will follow shortly thereafter [10, 11]. Geometric nonuniform fast Fourier transform codes [9, 15] and fast boundary integral solvers for elliptic systems [17, 16, 14] are in progress. The 3D surface tracking method described in [1, 2] has been implemented as part of the Berkeley Fluid Animation and Simulation Toolkit (BFAST), which has been open source released and is available on SourceForge.

References

[1] A. Bargteil, T. G. Goktekin, J. F. O'Brien, and J. Strain. A semi-Lagrangian contouring method for fluid simulation. *ACM Trans. Graphics*, 25:19–38, 2006.

[2] A. W. Bargteil, T. G. Goktekin, J. F. O'Brien, and J. A. Strain. A semi-Lagrangian contouring method for fluid simulation. *Visual Proceedings of ACM SIGGRAPH*, 2005.

[3] J. T. Beale and J. Strain. Locally-corrected semi-Lagrangian methods for Stokes flow with elastic interfaces. *J. Comput. Phys.*, 227:3896–3920, 2008.

[4] T. Chen and J. Strain. Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems. *J. Comput. Phys.*, 227:7503–7542, 2008.

[5] D. Farina, I. S. Sammis, and J. Strain. Fast Voronoi-based distance finding to piecewise-polynomial curves. Preprint, UC Berkeley Mathematics Department, 2011.

[6] J L Levenberg. *Accurate adaptive contour finding using C^1 data.* PhD thesis, University of California at Berkeley, 2003.

[7] J. L. Levenberg, I. S. Sammis, and J. Strain. Accurate adaptive contouring of C^1 surfaces. Preprint, UC Berkeley Mathematics Department, 2011.

[8] I S Sammis. *Implicit and high-order methods for moving interfaces*. PhD thesis, University of California at Berkeley, 2008.

[9] I. S. Sammis and J. Strain. A geometric non-uniform fast Fourier transform. *J. Comput. Phys.*, 228:7086–7108, 2009.

[10] I. S. Sammis and J. Strain. High-order semi-Lagrangian contouring for geometric moving interface problems. Preprint, UC Berkeley Mathematics Department, 2011.

[11] I. S. Sammis and J. Strain. Implicit semi-Lagrangian contouring for stiff geometric moving interface problems. Preprint, UC Berkeley Mathematics Department, 2011.

[12] J. Strain. A fast semi-Lagrangian contouring method for moving interfaces. *J. Comput. Phys.*, 169:321–342, 2001.

[13] J. Strain. Locally-corrected spectral methods and overdetermined elliptic systems. *Jour. Comput. Phys.*, 270:1–13, 2007.

[14] J. Strain. Fast ADI iteration for first-order elliptic systems. Preprint, UC Berkeley Mathematics Department, 2011.

[15] J. Strain. Geometric nonuniform fast Fourier transform by the butterfly algorithm. Preprint, UC Berkeley Mathematics Department, 2011.

[16] J. Strain. Locally-corrected spectral boundary integral methods for elliptic systems. Preprint, UC Berkeley Mathematics Department, 2011.

[17] J. Strain. Locally-corrected spectral volume potentials for elliptic systems. Preprint, UC Berkeley Mathematics Department, 2011.

4. Personnel Supported During Duration of Grant

John Strain	Professor, University of California, Berkeley
Tianbing Chen	Graduate Student, University of California, Berkeley
Ian Sammis	Graduate Student, University of California, Berkeley
Jed Duersch	Graduate Student, University of California, Berkeley
Daniel Greengard	Graduate Student, University of California, Berkeley
Darsh Ranjan	Graduate Student, University of California, Berkeley
Minhtri Nguyen	Undergraduate Student, University of California, Berkeley

Acknowledgment/Disclaimer This work was sponsored in part by the Air Force Office of Scientific Research, USAF, under grant number FA9550-08-1-0131. The views and conclusions expressed herein are those of the authors, and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the US Government.